(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 December 2001 (13.12.2001)

PCT

(10) International Publication Number
WO 01/95550 A2

(51) International Patent Classification⁷: H04L 7/00, H04J 3/06

(21) International Application Number: PCT/NO01/00234

(22) International Filing Date: 6 June 2001 (06.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20002884 6 June 2000 (06.06.2000) NO

(71) Applicant (for all designated States except US): ONTIME NETWORKS AS [NO/NO]; Orionveien 12, N-0489 Oslo (NO).

(72) Inventors; and
(75) Inventors/Applicants (for US only): HOLMEIDE, Øyvind [NO/NO]; Orionveien 12, N-0489 Oslo (NO). LILJESTRÖM, Lennart [SE/SE]; Bystammogatan 2, S-725 91 Västerås (SE).

(74) Agents: ANDERSEN, Bjørn et al.; Onsagers AS, P.O. Box 265 Sentrum, N-0103 Oslo (NO).

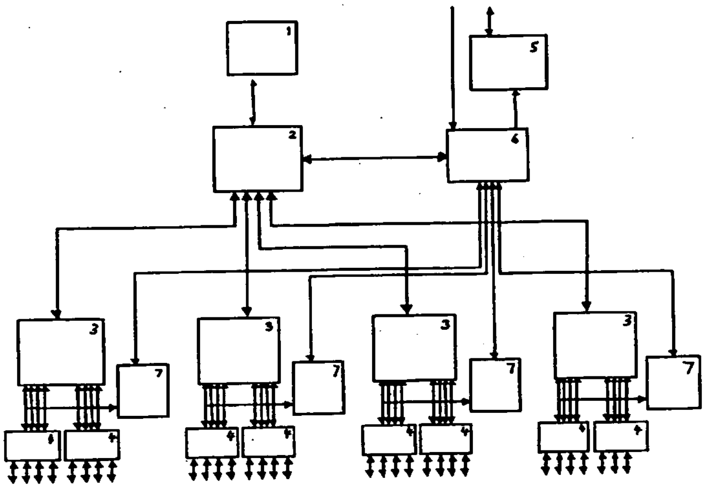
(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DISTRIBUTING TIME INFORMATION



(57) Abstract: An integrated network element and time server for distribution of time information in a computer network. The network element comprises means for routing data packets between communication ports, as well as a local clock and means for generating time update request packets and time update reply packets. The tight integration of communication services and time services enables the integrated network element and time server to distribute time update information while avoiding inaccuracies resulting from switching or routing operations. Also described is a method for synchronizing the local clocks of nodes in a communication network, taking advantage of the properties of the integrated network element and time server. The method provides automatic configuration of a time update structure in the network, so that time update information is distributed from one or more master time servers, with integrated network element and time servers working as slave time servers passing the time update information on throughout the network.

WO 01/95550 A2

Distributing time information

The invention concerns a method for time distributions services in a computer network as well as a integrated switch and time server and a computer network operating in accordance with the method.

When a plurality of computer systems are connected over a computer network it is often of great importance that the respective systems' local clocks are synchronized. This is particularly important in communication in substation automation applications in high and medium voltage power girds, where raw data is sent from producer to receiver nodes. Other examples include real time systems like process control systems where the respective parts of the system necessarily must operate on a common time reference.

In order to achieve this it has been common that communication in particularly critical systems has been based on expensive communication systems with separate lines, such as fiber optics, only used for the distribution of time information.

It is, however, also developed standards that aim to improve the exchange of time information over the same computer network as other data communication. Examples include RFC 1305 - Network Time Protocol (NTP) and RFC 2030 - Simple Network Time Protocol (SNTP). Briefly described, the latter is based on a client requesting a time update from a time server, where the time server receives timing signals from a very accurate source such as GPS (Global Positioning System) or an atomic clock. The time request being sent to the server includes a time stamp T1 indicating when the packet was sent from the client according to the client's local clock. As a reply to the request the server transmits a reply packet back to the client. The reply packet includes two time stamps, T2 and T3, where T2 is the time the request was received by the time server and T3 is the time the reply packet was sent from the time server. When the client receives the reply from the time server it determines a time of arrival, T4, for the reply. It can be seen that out of these four time stamps, T1 and T4 are based on the local clock of the client, while T2 and T3 are according to the local clock of the time server.

Based on these time stamps the client is able to calculate the following

$$d = (T4 - T1) - (T3 - T2) \qquad (1)$$

and

$$t = ((T2 - T1) + (T3 - T4)) / 2 \qquad (2)$$

where

d is the round trip delay, and

t is the local clock offset.

This method suffers from a number of weaknesses. The inclusion of additional network elements such as switches between the client and the server would add inaccuracies to the method. Other inaccuracies are related to the registration of the different time stamps. The latter are addressed in the same inventor's co-pending application entitled "Method for ensuring access to a transmission medium at a predetermined point in time and time server using the method". The problem introduced by the inclusion of additional network elements between the client and the server is the uncertainty in the round trip delay. Equation (2) is based on the assumption that the round trip delay is constant and equal in both directions. This assumption is justifiable as long as there is only one drop link and possibly passive hubs between the client and the server. If the reply and request packets have to wait in output buffers on any intervening switch, however, the delay is not constant and will normally not be the same in both directions. Because of this the round trip delay d will no longer be canceled out of equation (2), and inaccuracy increases.

US-patent 5,481,258 describes a distributed system, particularly a paging system, comprising a system controller and a plurality of distributed paging stations. The paging stations coordinate their respective clocks through timing information transmitted from the system controller. The system controller transmits a time mark, and at a future time it transmits the time mark send time. Each receiving paging station registers the time at which the time mark arrived according to its own local clock and they measure the time interval between the time at which the time mark arrived and the time at which the time mark was transmitted by the system controller. By subtracting the time at which the time mark was transmitted by the system controller and the propagation time to the respective paging station, each paging station can determine and correct the error in its own clock.

US-patent 4,815,110 describes a method for synchronizing clocks in a bus type local network, such as Ethernet. The method is based on letting one unit on the network operate as a master node. From the master node a synchronizing message is transmitted, addressed to all the nodes, including the master node. All the nodes then register the time at which the synchronizing message is received. Then the master node transmits a clock time message containing the master node clock state when it received the synchronizing message. The respective slave nodes will compare the received master clock state with the clock states which have been read in the slave nodes and correct their local clocks in accordance with the results of this comparison.

This method is advantageous if all the slave nodes are connected directly to the same bus or to a hub based Ethernet network. The only difference in time for the registration of the synchronizing message at the master node and at the slave nodes will be the difference in propagation delay. This difference will be static and may be corrected through calibration. None of the publications mentioned above addresses the problem introduced in a network structure based on switches. If there is only one switch between the master node and the slave nodes and the switch is based on so called "shared memory" architecture, the delay through the switch will at least be the same for the master node and all the slave nodes. However, if there are several switches between the master node and the slave nodes, or if the switch is not of the "shared memory" type, the delay will be variable.

It is therefor an object of the invention to provide a method for distributing time information on a computer network where the problems and disadvantages described are reduced.

In particular it is an object of the invention to provide a method for distributing time synchronization information in switched networks, without introducing errors due to transmission delay in network elements between time server and time client.

It is a further object of the invention to achieve distribution of time information in switched Ethernet with an accuracy that makes such networks a viable alternative to networks with fixed time slots. Profibus and MVB are examples of network implementations with fixed time slots. A relevant real time system that requires accurate time synchronization via its communication infrastructure is High Voltage (HV) substation automation system.

The advantage of real time systems with fixed time slots is that the arrival time of data packets is deterministic. This will not be the case in switched networks based on e.g. IEEE 802.3 (Ethernet) for the reasons described above. It is, however, possible to overcome this problem by making sure that each packet with measurement data contains an accurate time stamp indicating when the measurement was performed. The problem is then reduced to a problem of synchronization, where the challenge is to synchronize every node in the network where measurements are performed. It should however be pointed out that even though the invention primarily will be described in relation with the synchronization of substations over an Ethernet type network, the invention will be found useful in numerous other applications and in connection with other types of networks where it is desirable to maintain a high degree of synchronization of computers and other equipment connected to the network.

The stated objects of the invention are achieved through the features described in the independent claims. Advantageous embodiments and additional features are described in the dependent claims.

The invention will now be described in further detail by way of examples, and with reference to the accompanying drawings, where

Fig. 1 illustrates the exchange of information in a unicast SNTP time information distrubution,

Fig. 2 shows the message format according to SNTP version 4,

Fig. 3 shows a block diagram illustrating delay between a time server and a client with an intervening switch,

Fig. 4 illustrates a block diagram of an embodiment of an integrated time server and switch according to the present invention,

Fig. 5 shows a flow chart illustrating a first method for ensuring access to the communications port,

Fig. 6 shows a flow chart illustrating a second method for ensuring access to the communications port,

Fig. 7 shows a flow chart illustrating a third method for ensuring access to the communications port,

Fig. 8 shows a flow chart illustrating a fourth method for ensuring access to the communications port, and

Fig. 9 shows a network where the various nodes are synchronized in accordance with the invention.

The following examples of embodiments of the invention will be described with reference to the Simple Network Time Protocol. It should be pointed out, however, that although this is a convenient format for time information distribution, the invention is by no means limited to such implementations.

Figure 1a illustrates the flow of information in a unicast distribution of time information according to the SNTP protocol. Unicast means that a request for time update is addressed to one time server only, and only this time server replies. Alternatives to this are anycast, which is a broadcast request which is replied to by all time servers that receives the request, but the response is addressed only to the requesting time client, and multicast, which is a time update brodcast that is initiated by a time server.

At time T1 the client requests time information by sending a packet to the time server. The message format of this packet, which follows the IP and UDP headers, is shown in figure 2. All the fields of this message will not be described, reference is made to RFC 2030, Simple Network Time Protocol (SNTP) Version 4, for IPv4, IPv6 and OSI, by D. Mills, University of Delaware, October 1996.

The field denoted "Originate Timestamp" contains a time stamp indicating at which time T1 the request departed the client for the server in 64-bit time stamp format. When the request is received at the time server the time T2 is registered and recorded as a time stamp in the field denoted "Receive Timestamp". The message is then returned to the client, and the time T3 at which the reply departed the server for the client is recorded in the field called "Transmit Timestamp." When the reply is received at the client, the client records this time T4. Note that T1 and T4 will be according to the local clock of the client, while T2 and T3 are according to the local clock of the server.

As already described, the round trip delay is then calculated as

$$d = (T4 - T1) - (T3 - T2) \qquad (1)$$

and the local clock offset is

$$t = ((T2 - T1) + (T3 - T4)) / 2 \qquad (2)$$

It is also possible to calculate the time offset only based on either T1 and T2 or T3 and T4, but only if the propagation delay and any other delays can be compensated for through calibration or are negligible. Systems where timing information is transmitted or broadcast from a time server without any request from the client, such as the system described in the above mentioned US-patent 5,481,258 or SNTP in multicast mode, would correspond to the latter alternative.

Mention should also be made of the fields denoted "LI" (Leap Indicator), "Stratum" and "Precision". Leap Indicator is a two-bit code warning of an impending leap second to be inserted/deleted in the last minute of the current day. Stratum is an eight-bit unsigned integer indicating the stratum level of the local clock, such as "unspecified", "primary reference" and "secondary reference". Precision is an eight-bit signed integer indicating the precision of the local clock, in seconds to the nearest power of two. The values that normally appear in this field range from -6 for mains-frequency clocks to -20 for microsecond clocks found in some workstations.

If the time server is synchronized to a radio clock or other primary reference source and operating correctly, the LI field is set to 0 and the Stratum field is set to 1 (primary server); if not, the Stratum field is set to 0 and the LI field is set to 3. The Precision field is set to reflect the maximum reading error of the local clock. For all practical cases it is computed as the negative of the number of significant bits to the right of the decimal point in the NTP timestamp format. The value of these fields will then help the time client determine the quality of the reply from the time server. This will be described in more detail below.

Returning now to figure 1, figure 1b illustrates the problems introduced when round trip delay is not constant. In this case it takes much longer for the request to reach the server, than for the reply to reach the client. The total round trip delay can still be calculated according to equation (1), but the round trip delay will not be canceled out in the calculation of clock offset, and equation (2) is no longer accurate.

Reference is now made to figure 3, which illustrates the delays suffered by a time information packet transmitted from a time server over a switched network. The packet must first has to traverse the various protocol layers, here illustrated by an UDP layer, an IP layer, MAC layer and the IEEE 802.3 layer. The time it takes for the packet to traverse these layers is referred to as the transmit time, or T_{tx}. In addition the packet may have to wait in the output buffer of the port on which it is to be transmitted.

In this illustration the packet must pass through one switch where its address field is examined before it is routed to the proper output buffer or output buffers. The time it takes for the switch to examine the packet and the time spent in the output buffer may wary, and the total time spent in the switch will be referred to as T_{sw}. There is also a certain delay involved for the packet to propagate the transmission line or lines between the time server and the client. This is the propagation delay, T_{prop}. When the packet reaches the client it must again traverse the protocol layers in reverse order. This time delay is the receive time, or T_{rx}.

Propagation delay can be compensated for to the extent that it can be measured, since this delay is static. The transmit and receive times T_{tx}, T_{rx} can be dealt with through various measures. One way of reducing the receive time T_{rx} delay and improving accuracy is to detect the arrival of a time information packet, whether it is a request sent to the time server or a reply sent to the client, in hardware at the input port, before the packet traverses the various protocol layers. A method for eliminating transmit time T_{tx} by ensuring access to the transmission medium at the time corresponding to a time stamp pre set in the time information packet, instead of setting the time stamp before the packet traverses the protocol layers and the output queue, is described in the same applicant's co-pending application entitled

"Method for ensuring access to a transmission medium at a predetermined point in time and time server using the method".

The object of the present invention is to provide a method for synchronizing a communication network without suffering inaccuracies due to switches or other network nodes. This is achieved by integrating a time server in every network element that may cause variable delay when forwarding packets, and by following a time update scheme for the entire network as described below. In principle the network

In the example below, an integrated time server and switch will be described. It should be noted, however, that the invention is not limited to switches. A time server may be integrated in any bridge, gateway or computer on the network that receives and retransmits packets with variable delay.

Several architectures are used in the switches available on the market. Most common are shared memory, crossbar and hybrid crossbar. Shared memory depends on a central switch engine to provide high speed interconnection to all ports. Every packet must be examined in order to determine its routing. This approach may suffer from bottlenecks when traffic increases. The architecture is ideal for multicast and broadcast traffic. According to the crossbar architecture, every port is directly connected to a crossbar. The crossbar makes direct, high performance point-to-point connection between each port. This means good performance on unicast, but high complexity for multicast and broadcast traffic. This architecture is also not easily scaleable. Hybrid crossbar is based on a crossbar that provides connections between a set of switch engines each controlling e.g. eight local ports. The crossbar makes point-to-point connection when the destination address belongs to a port that is not local. The architecture is similar to pure crossbar designs, but more scaleable and flexible.

Figure 4 shows the block diagram of an integrated time server and switch according to a preferred embodiment of the invention. The switch is of the hybrid crossbar architecture and includes a switch CPU 1, controlling a crossbar 2, which again is connected to four Ethernet controllers 3, each with eight ports. The Ethernet controllers are connected to media independent interfaces MII 4 which connect the controllers 3 with Ethernet transceivers (not shown).

The time server comprises a time server CPU 5 which in the drawing is illustrated as a separate CPU, but which may also be the same CPU as the switch CPU. The time server CPU is connected to, and controls, a time stamp module 6. The time server CPU is also connected to a local clock (not shown) and, if the integrated switch/time server is operating as a master time server, as described below, preferably also to an external clock with great accuracy, such as a GPS receiver or an atomic clock. In the embodiment illustrated in figure 4, the time stamp module receives extremely accurate timing signals TTL (or Pulse Per Second - PPS) directly from the GPS, while the time server CPU 5 receives signals that shows the time in whole seconds. While the TTL signal is extremely accurate, the time given by the GPS packets received over RS232 are relatively inaccurate and not useful in real time critical applications.

The time stamp module 6 is connected to trigger modules 7, which monitor the input for SNTP requests, and upon detection of such a request generates a trigger signal that is sent to the time stamp module 6. The trigger modules 7 may also monitor the output for certain outgoing signals as described below. The time stamp module 6, when it is triggered by a trigger signal, generates a time stamp T2 or T4 indicating the time of arrival for an incoming packet. Time stamps T1, T3 for outgoing packets are also generated by the time stamp module 6, triggered either by the time server CPU 5 or by a trigger module 7, as will be described below.

Upon generation of time information packets, these must be transferred to the relevant output ports. This is illustrated by a connection between the time stamp module 6 and the crossbar 2, but it must be understood that several alternatives are possible, depending on such things as the particular architecture of the switch (or network element) and whether switch CPU 1 and time server CPU is implemented separately or as one single CPU.

In a preferable embodiment of the invention, the time server is configured to perform a method that will ensure access to the transmission medium at a given point in time. This is particularly useful for transmitting time information reply packets including a time stamp T3 (or T1) indicating when the packet was transmitted. In order to do this, the time stamp T3 (or T1) indicates a future time at which the packet will be transmitted, and the output port or ports on which the packet will be transmitted, will be disabled until this point in time. While the port or ports are disabled, the time information packet is queued at the relevant output ports as the first packet to be sent as soon as the output port again is enabled, e.g. by giving the packet highest priority according to IEEE 802.1p.

There are several ways of disabling the output ports. If the transmission medium allows full duplex transmission, so that there is no need to worry about incoming traffic, it is sufficient to ensure that no new packets will be transmitted after any ongoing transmission has been completed, and to set the time stamp T3 (or T1) equal to the point in time at which the output port or ports again will be available.

This is done as illustrated in figure 5 by disabling 101 the transmission of new packets on the port or ports on which the time information packet is to be transmitted and at the same time recording the time of the local clock in the time server. Any ongoing transmission of packets, however, is allowed to continue. The length of the period of time transmission is disabled, is set to be longer than the transmission time for a maximum size packet. In this way it is guaranteed that all ongoing transmission will be completed and all the output ports will be idle before the end of the disable period. Following this, the time of the local clock is read 102. Preferably the local clock is read at the same time transmission is disabled, but this may also be done at a later time, as long as the remaining time of the disable period is known. Based on the registered time of the local clock, a time stamp TS is generated 103. The Transmit Time stamp set equal to the time of the local clock recorded when transmission was disabled plus the length of the disable period.

A time information packet containing the time stamp is then generated and placed 104 in the first position of the output queues.

After a predetermined time 105, transmission is again enabled 106, and the time update packet will immediately be transmitted on the ports on which it is queued.

If, on the other hand, the transmission medium does not allow full duplex transmission, it is not sufficient to make sure that the output port or output ports on which the time packet is to be transmitted are not busy transmitting when it is time to transmit the time information packet. It is also necessary to make sure that no data will be received at this time. According to the invention, this is done by transmitting a signal on the output port or output ports prior to generating the time stamp and the time information packet.

Figure 6 describes a method according to the invention, where outgoing as well as incoming traffic on one port is disabled through the transmission of a dummy packet.

In order to disable transmission out of the time server and at the same prevent other nodes to send data on the transmission media, a dummy packet is generated and placed 201 in the output queue of the port on which it is to be transmitted. The dummy packet contains a particular pattern which makes it possible to identify.

After the dummy packet is placed in the output queue, the output port is monitored 202. As soon as the particular pattern that identifies the dummy packet is detected 203 as being transmitted on the port, the time T_0 of the local clock is read 204. The detection of the dummy packet will preferably be performed by the trigger module

7, which will send a trigger signal to the time stamp module 6. Following this, preferably upon receipt of the trigger signal by the time stamp module 6, a time stamp TS is generated 205 and registered as the Transmit Time Stamp of a time information packet. The time stamp is set equal to the time of the local clock recorded when transmission was disabled plus the time it takes to complete transmission of the dummy packet, plus the length of the minimum allowable time gap between packets, or $TS = T_0 + T_{flush} + T_{gap}$. By placing the dummy packet in the output buffer 206 making sure that no packet is queued between the dummy packet and the time information packet, the time information packet will by necessity be transmitted when the time of the local clock reaches the same value as the time stamp in the Transmit Time Stamp field of the time information packet.

Referring now to figure 7, an alternative embodiment will be described where the dummy packet is replaced by a busy signal. According to the Ethernet standard, such a busy signal is referred to as a back pressure signal and is normally used to prevent incoming information when the output buffers of a switch is about to overflow.

The time server enables 301 a busy signal on the output port on which the time information packet will be transmitted. At the same time the local clock time T_0 is registered 302. Alternatively the local clock time is registered when the busy signal is detected on the output port by the trigger module 4 which sends a trigger signal to the time stamp module. Following this, a time stamp TS is generated 303. The time stamp TS is set to equal the registered time of the local clock when the busy signal was enabled or detected, T_0, plus the time the busy signal will remain enabled or the time it will remain enabled after having been detected, plus the minimum allowable time gap between packets, or $TS = T_0 + T_{busy} + T_{gap}$.

The time information packet is then placed 304 in the first position of the output queue where it waits 305 until the busy signal is disabled 306. This will ensure that the time information packet is transmitted when the time of the local clock equals the time stamp in the time information packet.

It must be noted that while the first embodiment described with reference to figure 5 can be utilized also when the time information packet is to be transmitted on several output ports at the same time, this is not the case with the two embodiments where access to the transmission media is ensured through the transmission of a dummy packet and a busy signal respectively. This is because the time of the local clock is registered and the transmission of the dummy packet or the busy signal is started as soon as the output port is available. When the transmission of the time information packet will take place on several output ports, however, there's no reason to expect that these ports will become available at the same time.

Referring now to figure 8, an alternative embodiment will be described, which enables the transmission of time information packets on multiple ports. As in the embodiment described with reference to figure 7, the time server enables 401 a busy signal, but this time on a plurality of output ports. There are then three preferable alternatives for when the time T_0 of the local clock is read 402. It can either be read at the time tne busy signal is enabled, as soon as the busy signal is detected on at least one output port, or as soon as the busy signal is detected on all the output ports. The detection of busy signals on the output ports will be performed by the trigger modules 7 which will send a trigger signal to the time stamp module 6 upon detection of the busy signal. It is obviously possible to select any other point in time within the interval between enabling the busy signal and actually generating the time information packet, as long as the time from reading T_0 and until transmitting the time information packet is deterministic. The three alternatives mentioned are well defined, however, and are therefore the preferred alternatives.

As soon as the time T_0 is read, a time stamp is generated 403. The time stamp is again given as $TS = T_0 + T_{busy} + T_{gap}$.

If T_0 is registered when the busy signal is enabled or when it is detected on at least one port, the busy signal should remain enabled at least for a period of time T_{busy} that is as long as the time it takes to transmit a maximum size packet, or $T_{busy} \geq T_{flush}$. This will ensure that ongoing transmission of packets will be completed and the transmission of the busy signal will have started on all the ports before the busy signals are disabled and the time information packet is transmitted. If the time of the local clock T_0 is only registered after the busy signal has been detected on all the output ports, it is sufficient that the busy signals remain enabled only so long that the time server has sufficient time to generate the time information packets and place them in the first position of the output queues. The exact sequence of these steps are not critical, however, as long as the remaining time during which the busy signal is known at the time the local clock is read. Figure 8 illustrates an embodiment where following the detection 404 of the busy signal on all the output ports, the time information packets are placed 405 in the output buffer. After the preset period of time has elapsed 406, the busy signal is disabled 407 simultaneously on all the output ports.

The embodiments just described, which ensures that the transmission medium can be accessed at the point in time at which the time stamp in a packet is set, is further described in the same applicant's co-pending application entitled "Method for ensuring access to a transmission medium at a predetermined point in time and time server using the method".

It should be noted that in the embodiments described above, the time stamps of outgoing packets will be generated by the time stamp module 6 when this is triggered. If the time stamp is generated on basis of a detected dummy packet or busy signal on one or all of the output ports, the trigger signal will be emitted by the respective trigger modules 7. If, on the other hand, the time stamp is generated on basis of the point in time at which the output port or ports are disabled or the busy signal enabled, the trigger signal will be emitted by the time server CPU 5.

It should also be noted that in the embodiments described in relation to figure 6, figure 7 and figure 8, respectively, as they relate to half duplex transmission, involve a certain risk of collision. This may occur because following the end of transmission of the dummy packet or the busy signal, respectively, the time information packet must wait a period of time T_{gap} before it can be transmitted. T_{gap} is equal to the Inter Packet Gap (IPG), which is also the period of time a receiving client must wait following the end of reception of the dummy packet or busy signal. It is therefore possible that the client will begin transmission before or at the exact time when it receives the time information packet, resulting in a collision. Numerous methods for collision detection are known in the art. It is preferable that such a method is implemented in the time server and that upon detection of a collision, the process is restarted, including the transmission of a new dummy packet or busy signal and the generation of a new time information packet. The exact method selected for collision detection is dependent on factors such as the nature of the transmission medium, and is not part of this invention.

Figure 9 illustrates a computer network where network elements are synchronized by a method according to the present invention. The various nodes are shown as switches and computers, but other types of network elements may also be included, such as bridges, routers, gateways and hubs. It will, however, normally not be desirable to integrate time servers in network elements with static delay.

In figure 9, one switch is functioning as a master time server 10. The master time server 10 is connected to an external clock 11, such as a GPS receiver, from which it receives highly accurate timing signals. The master time server 10 at all time keeps its own local clock synchronized with the external clock 11. Further the master time server 10 is connected to a number of switches 12, 13, 14 that function as slave time servers. This means that the slave time servers 12, 13, 14 are time clients with respect to the master time server 10, but function as time servers with respect to any network node connected to them but not to the master time server 10. In the example of the drawing it can be seen that one time server 12 is connected to another switch 15, one time server 13 is connected to a personal computer 20, while one time server 14 is connected to an additional switch 16 as well as a personal computer 21. The personal computers 20, 21 are end nodes of the network and have time clients running on them, receiving their update from the respective slave time servers 13, 14. The additional switches 15, 16 have time clients that receive their time update information from the slave time servers 12, 14 they are connected to, while they act as time servers with respect to additional nodes to which they are connected, in this case the personal computers 22, 23, and possibly each other, since they are interconnected.

It should be noted that while the master time server 10 in the example of figure 9 is a switch, it could alternatively be implemented as running on a computer.

Several alternatives can be implemented in order to configure the respective switches regarding whether or not they are master or slave time servers, and in the case a switch functions as a slave time server, on which drop links (or communications ports) it functions as a time server and on which drop links it functions as a time client. The most straight forward way of handling this is to configure each switch manually, in software or in hardware, e.g. by setting dip switches for each communications port. In this case the master time server 10 would be acting as a time server on all its regular communications ports, while the slave time servers 12,...,16 would be set to act as time clients on the ports that are connected to a switch that is closer to the master time server and as time servers on ports connected to a switch that is further removed from the master time server 10 (in terms of number of hops) or to an end node 20,...,23. The ports that interconnect two switches 15, 16 could be set the same (e.g. either both server only or both client only), which would effectively disable any update between these, or one could be set as server and one as slave in order to achieve redundancy in the system. Following configuration the time clients (i.e. the end nodes 20,..., 23 and the time client function of the slave time servers) will regularly transmit unicast requests on the drop links that are so configured, e.g. every 10 seconds. If more than one port/drop link is set to transmit requests, the requests could be transmitted on all these ports, to achieve redundancy, or to only one in order to reduce network traffic.

It is, however, desirable to achieve automatic configuration of the switch/time server nodes, and this is achieved through the following preferable embodiment of the invention. Whenever a switch/time server is connected to the network, it will immediately transmit an anycast signal on all its ports. An anycast signal according to SNTP is a request for a time update response from any time server that receives the request. The request is a broadcast signal, which means that it will reach every node on the network. The nodes that do not contain a time server, such as the personal computers 20,..., 23 according to this example, will obviously not reply to the anycast signal. Neither will any time server that has lost its synchronization by not having received a time update signal from any time server closer to the master time server within a defined time, e.g. within the last 10 seconds. All the synchronized time servers will reply, however.

When the newly connected switch/time server receives the replies to its anycast request, it will determine the round trip delay for all the responses according to equation (1). The round trip delay will be significantly shorter for replies from switches that are connected to the newly connected switch over one drop link only, hence the switch will be able to rule out any time servers that are connected via several drop links. Valid time servers for the newly connected switch will then be time servers with which it is immediately interconnected through one drop link, while it may still be able to reply to time update requests received on any port.

Note that if time server 12 is connected to the network of figure 9 while the master time server 10 and the slave time server 15 are both up and running, it will determine that both of these are reached over one drop link only. This means that an update signal from slave time server 15 might be preferred before an update signal from the master time server 10, even though the local clock of slave time server 15 is a result of updates from time server 14, which again is updated from master time server 10. However, as described with reference to figure 2, the time client can determine the quality of the time reference signal based on some of the fields of the time update packet, and this would make the master time server 10 the preferred time server for the switch/slave time server 12.

According to a preferred embodiment of the invention, the determination of quality of the time reference signal is based on the distance between the respective time server and the closest master time server. In general this means that in the time information packets generated by any master time server, this particular quality information value is set equal to a particular reference value, and in the time information packets generated by any slave time server, this quality information value will be set equal to the quality information value of the latest time information packet used to update the respective slave time servers clock increased by a certain increment value. In other words, if the quality information value of the master time server is one and the increment for each step away from the master time server is one, time information packets from the master server will include a quality value of one, time information packets generated by slave time servers that are directly connected to the master time server and updated by it will include a value of two, slave time servers that are not directly connected to the master time server, but are being updated by one of the time servers that are will include a value of three, and so on throughout the network.

This can be implemented fairly straightforward if SNTP is the protocol being used. The field referred to as Stratum in the SNTP specification is utilized in order to achieve the mentioned quality ranking of the respective time servers. According to SNTP, Stratum is an eight-bit unsigned integer indicating the stratum level of the local clock, with values defined as follows:

Stratum	Meaning
0	unspecified or unavailable
1	primary reference
2 - 15	secondary reference
16 - 255	reserved

Time information packets transmitted form a master time server would have a stratum value of 1, while the slave time servers would transmit time information packets with a stratum value one higher than the stratum value of the time information packets used to update this slave time server. In other words, slave time servers updated from a master time server would have stratum values of two, servers updated from these would have stratum values of three, and so on.

In this way a self configuring switch/time server would be able to find several valid time servers from which it can be updated, and determine a list of priority between these. This redundancy means that if the preferred time server should become unavailable, the port on which it is connected will be reset. During operation the slave time server, acting as a time client, will send unicast requests at regular intervals, e.g. every 10 seconds. The unicast requests may be sent to the time server with highest priority only, or to all the time servers from which the best reply is chosen each time. If stratum is used to determine priority, the time server with the lowest stratum value will be preferred. In case of equal Stratum value from two or more connected time servers, round trip delay or other factors can be used as a tie breaker.

If a time server with a Stratum value of three suddenly loses its connection with the time server with Stratum value two, from which it has been updated, it must start receiving updates from a time server lower on its list of priorities. If the next eligible time server also has a Stratum value of three, the time server in question must change its Stratum value to four.

It will be noted that in this way, a situation where two time servers update each other will not occur. If the two servers have the same Stratum value, they must both either be master time servers or they must both be connected to a time server with Stratum one lower, and both will prioritize updates from other time servers than each other. If they do not have the same Stratum value, the one with the lower Stratum value must either be a master time server or it must be connected to a time server with Stratum value one lower than itself. In the latter case it will prioritize updates from the time server with Stratum lower than itself, not from the server with stratum higher than itself. If the network is reconfigured due to a failure of one or more nodes or drop links, the Stratum values will change accordingly and the description given above will still hold true.

If a switch/time server does not receive any time update replies within a certain period of time, e.g. 10 seconds, it can reset itself so as not to reply to anycast requests, but it may still continue updating its time clients by responding to unicast requests. If the situation continues and the switch/time server does not receive time update packets within a period of time such as e.g. 24 hours, it may discontinue its operation as a time server all together, and only transmit anycast requests until it again can become configured.

If a part of the network becomes isolated from all the master time servers, the Stratum value of the respective time servers in this part of the network will in theory be infinite, since there is no path to a master time server with Stratum value one. What will happen is that for a transitional period the time servers will increase their own Stratum value each time they request a time update, until a maximum value is reached. This can be illustrated by an example where two interconnected time servers are disconnected from the rest of the network at a time where one has a Stratum value of 5 and the other has a Stratum value of 6. The first change will happen when the time server with Stratum value of 5 realizes it can only be updated from the time server with Stratum value 6. It will then change its own Stratum value to 7. Following this the other time server will be updated by the first time server and increase its Stratum value to 8. The switches/servers will then update each other until a maximum value is reached. According to SNTP this absolute value is 15, but other implementations may be chosen within the scope of the invention. When this value is reached, the time servers determine that they do no longer receive valid time update information. According to a preferred embodiment of the invention, the time servers will then transmit anycast requests and receive updates from all available time servers. The local clock will then be adjusted to an average of local clock and all responses received from time servers connected only through one drop link (as determined by the calculation of round trip delay). In this way the drifting of the various clocks in the time servers of the isolated part of the network will presumably cancel each other out to a certain extent.

Alternatively, a switch/time server may set its Stratum value to a pre defined value such as zero if it does not receive any valid time update within a certain period of time, or if its Stratum value reaches a certain threshold. According to one embodiment of the invention, a switch/time server will then switch to anycast requests as described above when its Stratum value is set to this pre defined value and/or it only receives time update packets with a Stratum value equal to this value.

Because of the tight integration of the switch and the time server, the time server will always have information available regarding whether the respective communication ports on the switch are in contact with the equipment at the other end of the drop link. In other words, the time server will "know" which of the acceptable time servers are available at any given time. This will greatly facilitate the dynamic reconfiguration of the network if any time server malfunctions or is added to the network. The affected time servers will immediately be able to go through the necessary steps in order to update their list of available time servers.

Within the scope of the invention it is possible to implement other ways of configuring the slave time servers, particularly if other time distribution protocols than NTP/SNTP are used, and the examples above must not be understood as limiting.

It should be noted that while the embodiments described above primarily refer to the invention being used in relation with time servers, the method claimed in the attached claims are directed towards a method for including a time stamp in a data packet in general, and that this data packet does not have to be a time information packet as such. It should further be noted that while the examples often refer to the Ethernet protocol, this too is by way of example, and it must be understood that a person with skill in the art to which the invention pertains, will see that the invention has similar applicability to any other similar communication protocol.

PATENT CLAIMS

1. Network element with a plurality of communication ports for receiving and transmitting data packets, means for routing data packets received at one port to one or more other ports for transmission, and a local clock, characterized in comprising an integrated time server with means for, upon receipt of a data packet indicating a time update request,
- reading the time of the local clock,
- generating a time information packet, said packet including a time stamp based on the registered time of the local clock and indicating the point in time at which the time information packet will be transmitted, and
- transmitting the time information packet as a reply to the time update request;
and an integrated time client with means for generating and transmitting data packets indicating time update requests and means for adjusting the local clock based on the time stamp of a time information packet received in response to such a time update request.

2. Network element according to claim 1, characterized in further comprising means in the time client for, following receipt of more than one time information packet, determining, based on round trip delay, which of the responding time servers from which said packets originated are connected over only one drop link, and ranking these time servers based on quality information found in the time information packets.

3. Network element according to claim 1, characterized in further comprising input means for receiving time signals from an external clock, and means for adjusting the local clock in accordance with said time signals.

4. Network element according to claim 1, characterized in that the means for routing data packets include a switch CPU (1) and a crossbar (2).

5. Network element according to claim 1, characterized in that the means for routing data packets include a switch CPU (1) and a shared memory.

6. Network element according to claim 1, characterized in that the means for routing data packets include a switch CPU (1), a hybrid crossbar (2) and Ethernet controllers (3).

7. Network element according to claim 1, characterized in that the means for generating a time information packet include a time server CPU (5), and a time stamp module (6) controlled by the time server CPU (5).

8. Network element according to claim 7, characterized in that the means for generating a time information packet further include trigger modules (7) each set to monitor respective communication ports and upon detection of a predetermined signal or data pattern, to send a trigger signal to the time stamp module (6).

9. Network element according to claim 1, characterized in that the means for transmitting a time information packet include means for making the communication ports on which time information packets are to be transmitted, unavailable for any communication that is not already ongoing and means for preparing the time information packets for transmission immediately upon termination of the period of time during which the communication port is unavailable.

10. Network element according to claim 1, characterized in that that the means for transmitting a time information packet include means for making the communication ports on which time information packets are to be transmitted, unavailable for any communication that is not already ongoing by generating a dummy packet containing a predetermined pattern, preparing said dummy packet for transmission immediately upon termination of any ongoing transmission,

means for detecting the predetermined pattern on the communication ports, and

means for generating the time information packet based on the time of the local clock when the predetermined pattern is detected, and preparing the time information packet for transmission immediately subsequent to the dummy packet.

11. Network element according to claim 1, characterized in that the means for transmitting a time information packet include means for making the communication ports on which time information packets are to be transmitted, unavailable for any communication that is not already ongoing by generating a busy signal that that will be transmitted on the output ports as soon as any ongoing transmission or reception on the respective port is completed and that will continue for a predetermined period of time,

means for detecting the busy signal on the communication ports, and

means for generating the time information packet based on the time of the local clock when the predetermined pattern is detected, and preparing the time information packet for transmission immediately subsequent to termination of the busy signal.

12. Network element according to any one of the previous claims, characterized in that said routing means include a switch CPU (1), said means for generating a time information packet include a time server CPU (5), wherein said switch CPU (1) and said time server CPU (5) are implemented as one unit.

13. Method for synchronizing the local clocks of the nodes in a communication network where at least one node is chosen as a master time server, and where nodes that are to be synchronized include a time client, said network comprising one or more nodes that are network elements with a plurality of communication ports for receiving and transmitting data packets and means for routing data packets received at one port to one or more other ports for transmission,
characterized in
- enabling at least a subset of said network elements to also operate as combined time client and slave time server;
- upon connection of a node including a time client to the network, broadcasting a time update request on the network;
- upon receipt of time information packets from time servers connected to the network as a response to a broadcast time update request, determining, based on a calculation of round trip delay, which of the responding time servers are connected only over one drop link,
- ranking these time servers based on quality information found in the time information packets, and creating a list of preferred time servers;
- upon receipt of a time update request in any node including a master time server or a slave time server
-- reading the time of the local clock,
-- generating a time information packet including a time stamp based on the registered time of the local clock and indicating the point in time at which the time information packet will be transmitted, and
-- transmitting the time information packet as a reply to the time update request; and
- at regular intervals, sending time update requests from the respective time clients to individual time servers selected from the respective time client's list of preferred time servers.

14. Method according to claim 13, characterized in that the quality information found in the time information packet indicates whether the time information packet was generated by

a master time server, or if not, how far removed the slave time server that generated the time information packet is from the nearest master time server.

15. Method according to claim 14,
characterized in that said quality information is generated through the steps of

- in the time information packets generated by any master time server, setting this quality information value equal to a particular reference value, and
- in the time information packets generated by any slave time server, setting this quality information value equal to the quality information value of the latest time information packet used to update the respective slave time servers clock increased by a certain increment value.

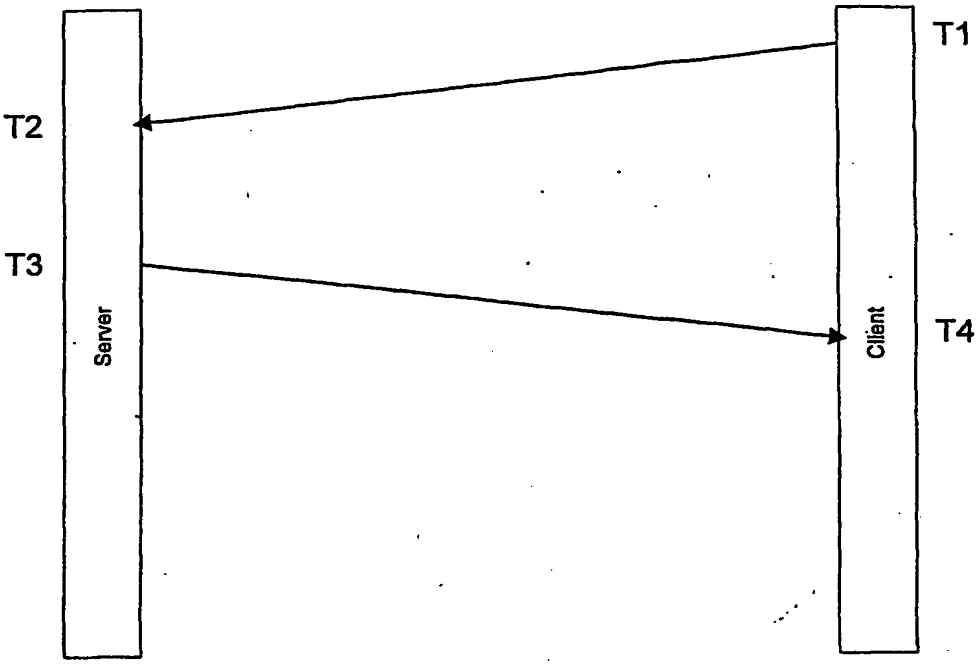


Fig. 1a

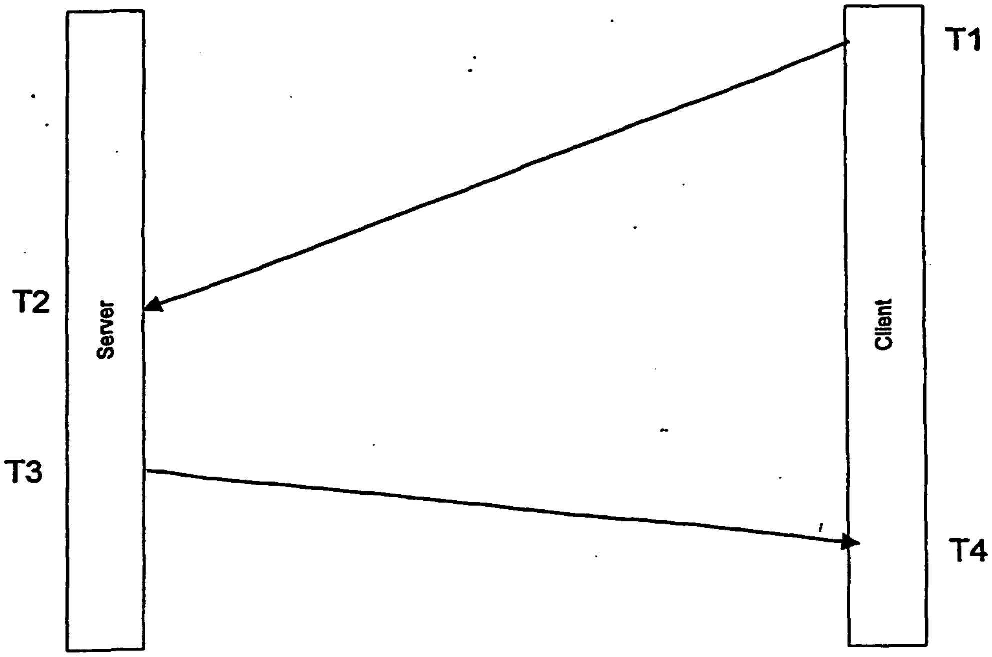


Fig. 1b

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|LI | VN  |Mode |    Stratum    |     Poll      |   Precision   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Root Delay                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Root Dispersion                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Reference Identifier                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   Reference Timestamp (64)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   Originate Timestamp (64)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Receive Timestamp (64)                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Transmit Timestamp (64)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Key Identifier (optional) (32)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                 Message Digest (optional) (128)               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 2

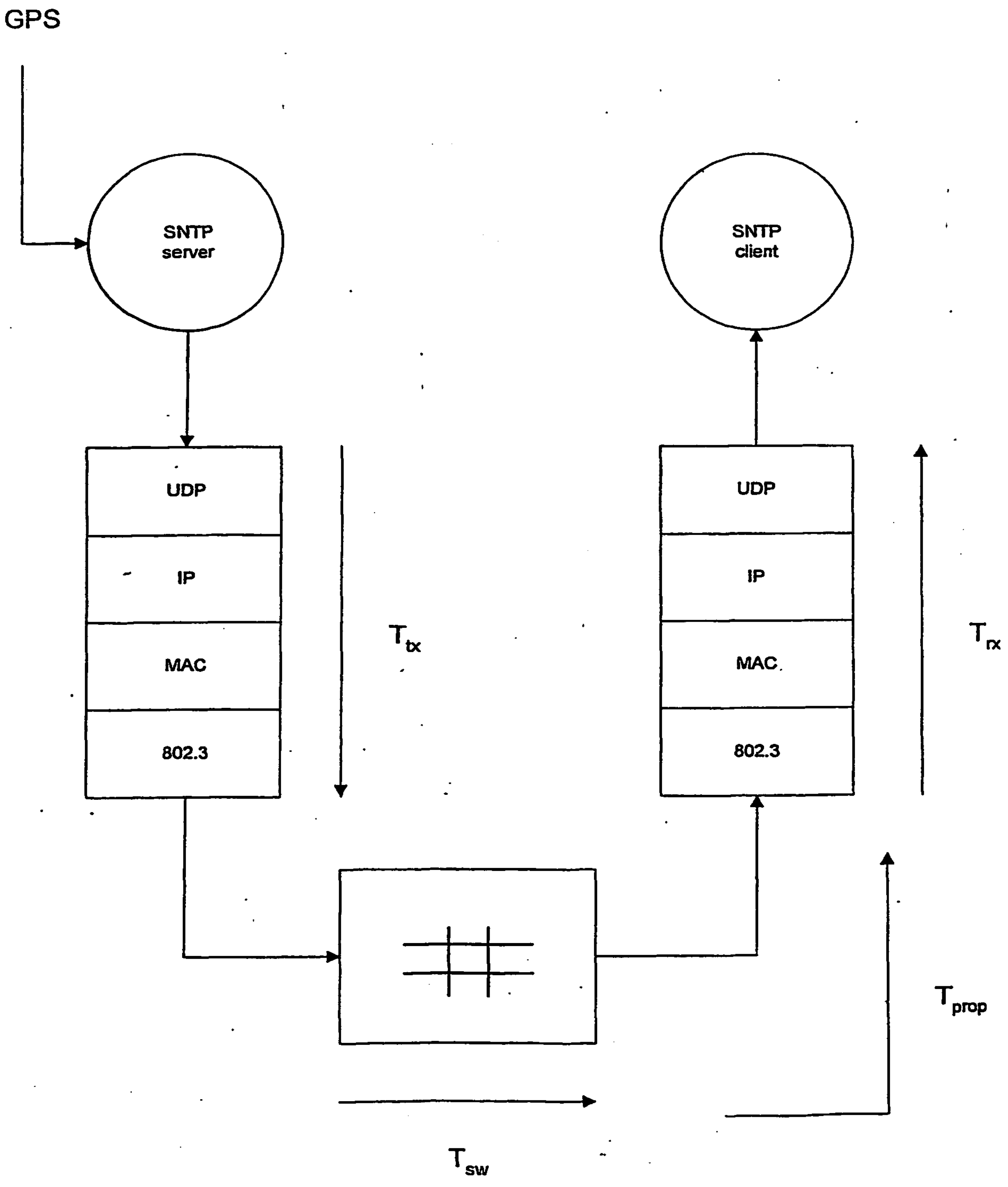

GPS
SNTP server
SNTP client
UDP
IP
MAC
802.3
UDP
IP
MAC
802.3
T_{tx}
T_{rx}
T_{prop}
T_{sw}

Fig. 3

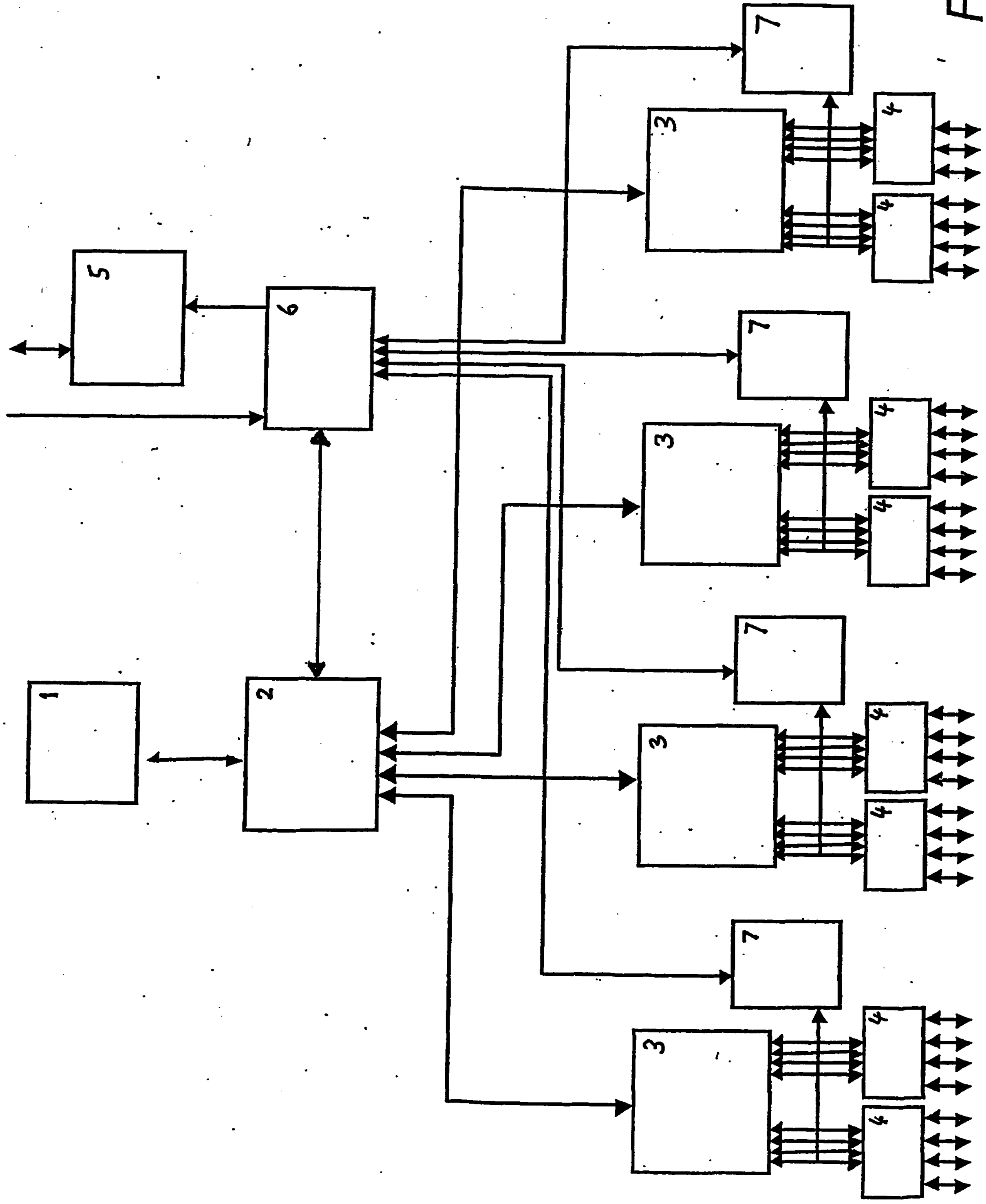


Fig. 4

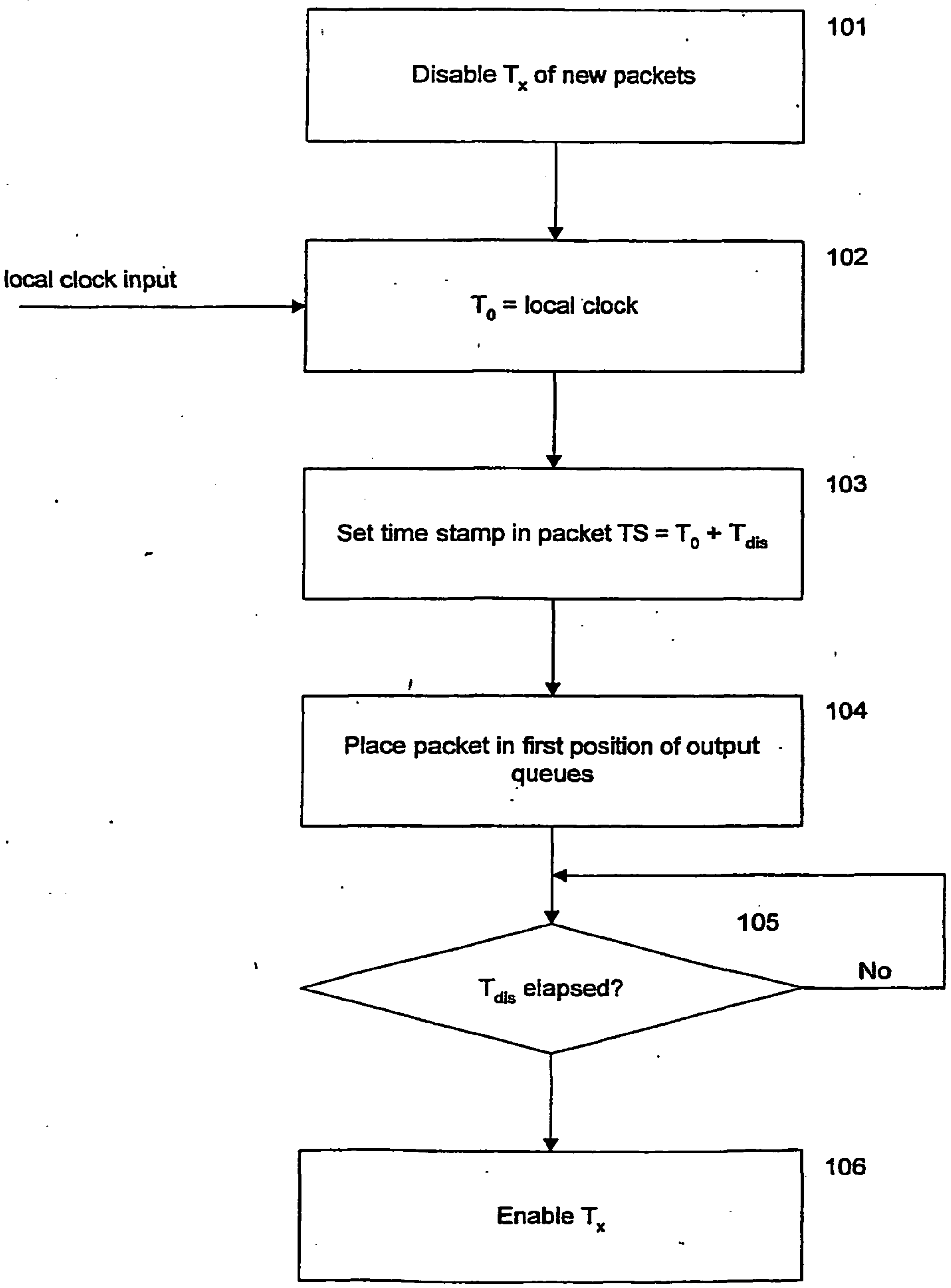

101
Disable T_x of new packets
102
local clock input
T_0 = local clock
103
Set time stamp in packet TS = $T_0 + T_{dis}$
104
Place packet in first position of output queues
105
T_{dis} elapsed?
No
106
Enable T_x

Fig. 5

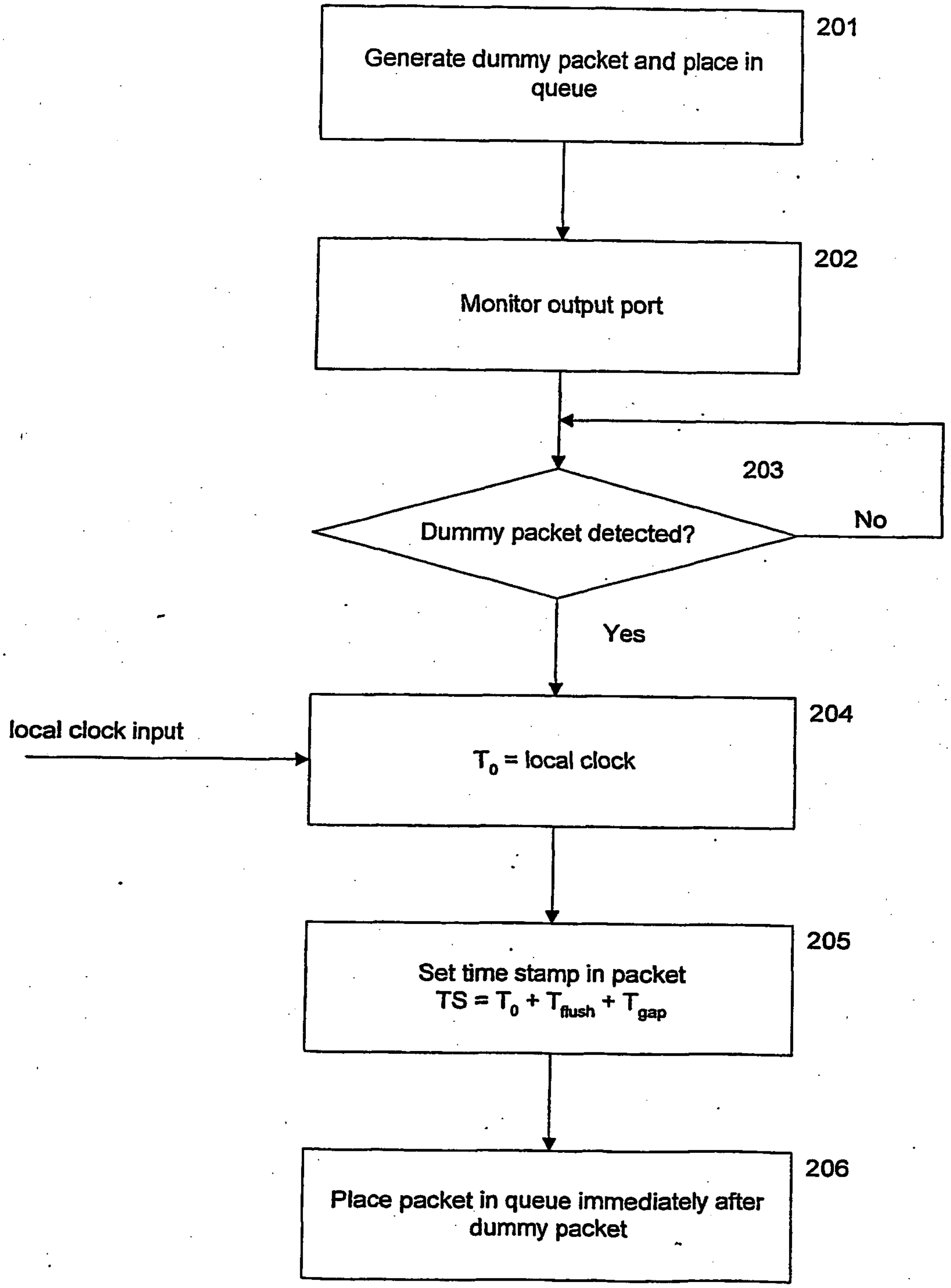

Generate dummy packet and place in queue
201
Monitor output port
202
203
Dummy packet detected?
No
Yes
local clock input
T_0 = local clock
204
Set time stamp in packet
$TS = T_0 + T_{flush} + T_{gap}$
205
Place packet in queue immediately after dummy packet
206

Fig. 6

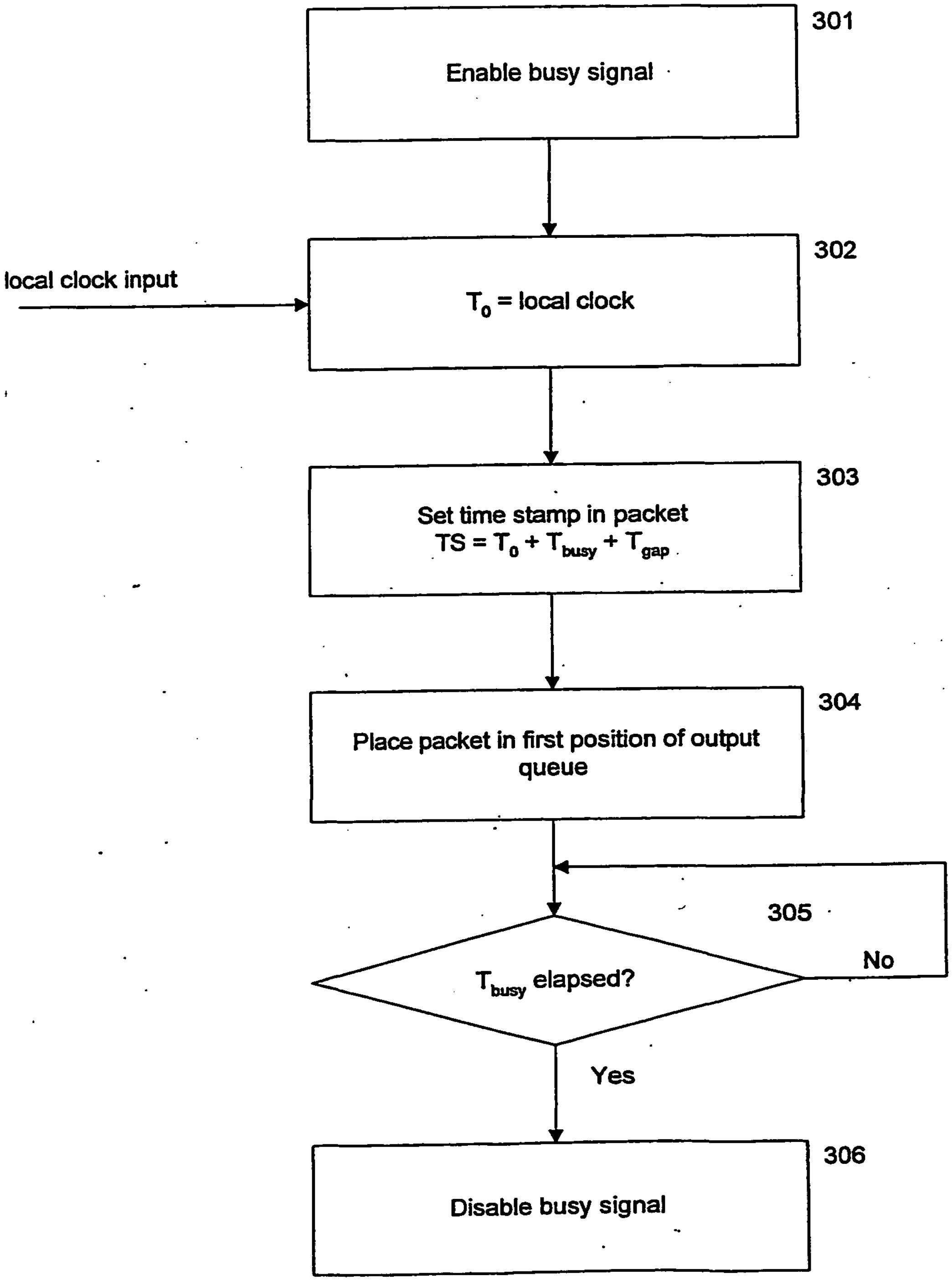

Enable busy signal
301
local clock input
T_0 = local clock
302
Set time stamp in packet
TS = $T_0 + T_{busy} + T_{gap}$
303
Place packet in first position of output queue
304
305
T_{busy} elapsed?
No
Yes
Disable busy signal
306

Fig. 7

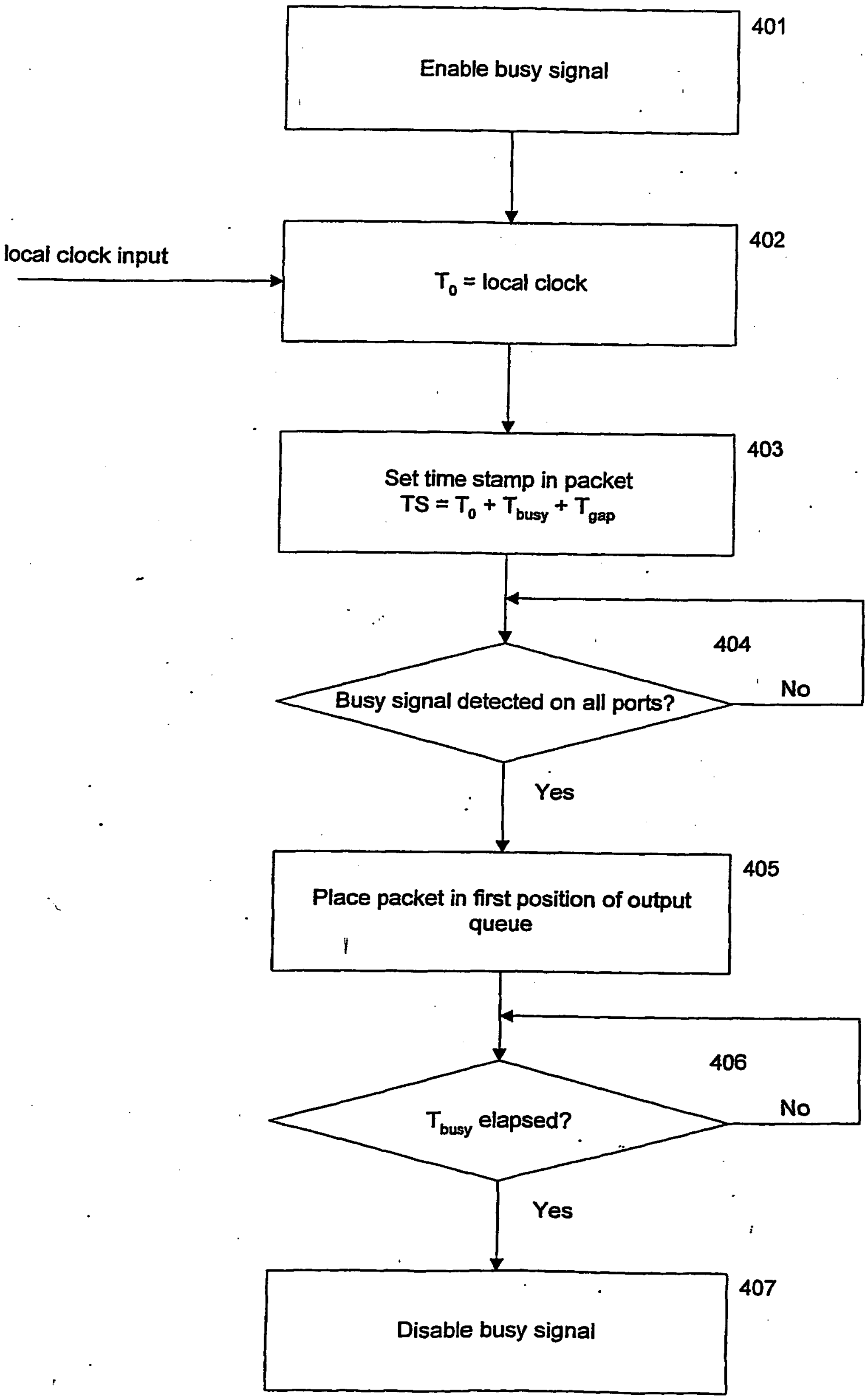

401
Enable busy signal
local clock input
402
T_0 = local clock
403
Set time stamp in packet
TS = T_0 + T_{busy} + T_{gap}
404
Busy signal detected on all ports?
No
Yes
405
Place packet in first position of output queue
406
T_{busy} elapsed?
No
Yes
407
Disable busy signal

Fig. 8

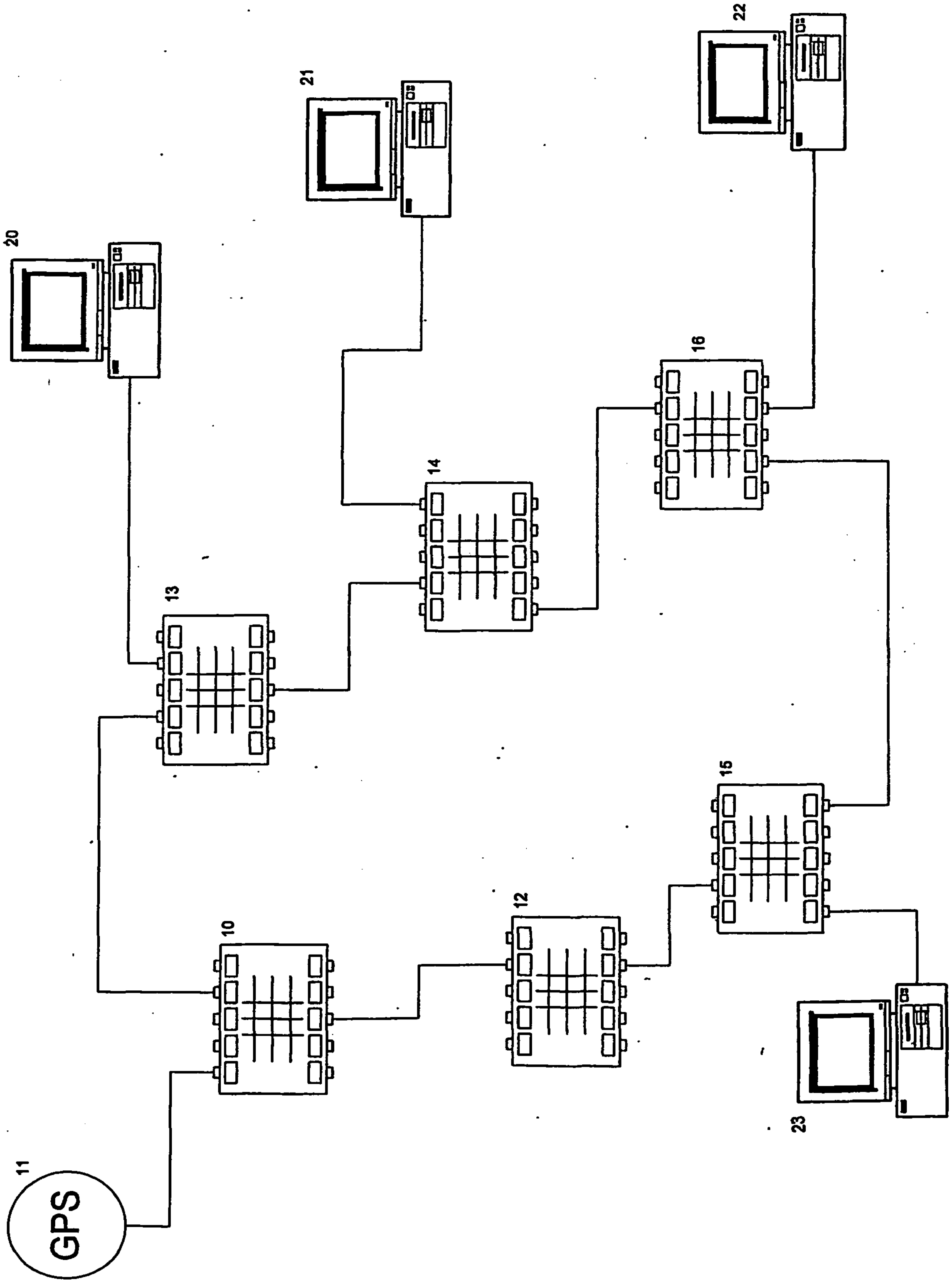


Fig. 9

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 December 2001 (13.12.2001)

PCT

(10) International Publication Number
WO 01/95550 A3

(51) International Patent Classification⁷: H04L 7/00, H04J 3/06

(21) International Application Number: PCT/NO01/00234

(22) International Filing Date: 6 June 2001 (06.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20002884 6 June 2000 (06.06.2000) NO

(71) Applicant *(for all designated States except US)*: **ONTIME NETWORKS AS** [NO/NO]: Orionveien 12, N-0489 Oslo (NO).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **HOLMEIDE, Øyvind** [NO/NO]: Orionveien 12, N-0489 Oslo (NO). **LILJESTRÖM, Lennart** [SE/SE]: Bystammogatan 2, S-725 91 Västerås (SE).

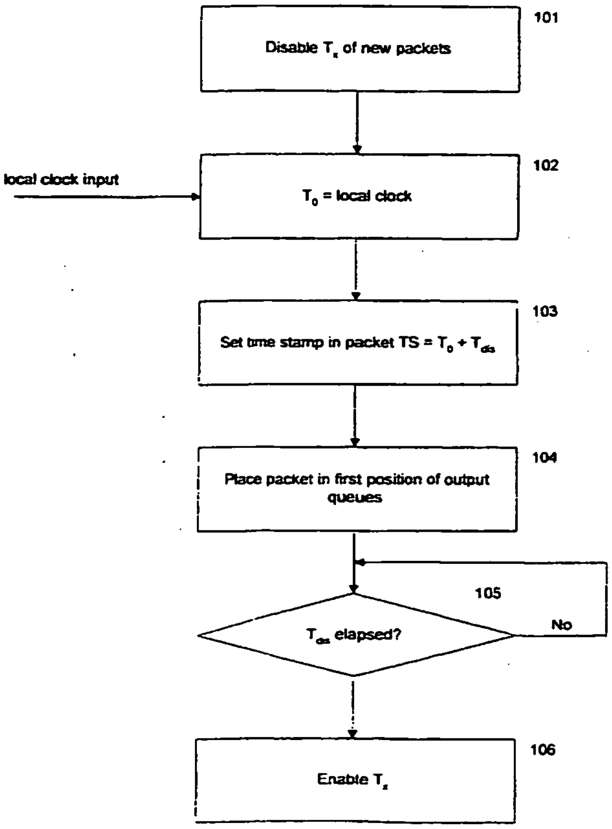
(74) Agents: **ANDERSEN, Bjørn** et al.: Onsagers AS, P.O. Box 265 Sentrum, N-0103 Oslo (NO).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: DISTRIBUTING TIME INFORMATION IN A COMMUNICATION NETWORK


101
Disable T_x of new packets
local clock input
102
T_0 = local clock
103
Set time stamp in packet $TS = T_0 + T_{ds}$
104
Place packet in first position of output queues
105
T_{ds} elapsed?
No
106
Enable T_x

(57) Abstract: An integrated network element and time server for distribution of time information in a computer network. The network element comprises means for routing data packets between communication ports, as well as a local clock and means for generating time update request packets and time update reply packets. The tight integration of communication services and time services enables the integrated network element and time server to distribute time update information while avoiding inaccuracies resulting from switching or routing operations. Also described is a method for synchronizing the local clocks of nodes in a communication network, taking advantage of the properties of the integrated network element and time server. The method provides automatic configuration of a time update structure in the network, so that time update information is distributed from one or more master time servers, with integrated network element and time servers working as slave time servers passing the time update information on throughout the network.

WO 01/95550 A3

Published:
— *with international search report*

(88) Date of publication of the international search report:
2 May 2002

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

I' ERNATIONAL SEARCH REPORT

Int...ational Application No
PCT/NO 01/00234

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L7/00 H04J3/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	D. MILLS: "Network time synchronization" MILLS, [Online] October 1989 (1989-10), pages 1-27, XP002902017 Retrieved from the Internet: <URL:http://www.eecis.udel.edu/mills/database/rfc/rfc1129/rfc1129b.pdf> [retrieved on 2001-10-23] the whole document ---	1-7, 12-15
A	"Application Note //23: Precise synchronization of computer networks: network time protocol (NTP) for TCP/IP" TRUETIME.INC., [Online] 5 July 1997 (1997-07-05), XP002902018 Retrieved from the Internet: <URL:http://www.truetime.com/DOCSn/ap23.pdf> [retrieved on 2001-10-24] the whole document --- -/--	1-7, 12-15

☒ Further documents are listed in the continuation of box C.

□ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 29 October 2001	Date of mailing of the international search report 21 12 2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Ismar Hadziefendic

2

INTERNATIONAL SEARCH REPORT

International Application No
PCT/NO 01/00234

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	NTP SYSTEM IMPLEMENTATION MODEL, [Online] 29 September 1997 (1997-09-29), XP002902019 Retrieved from the Internet: <URL:http://support.baynetworks.com/library/tpubs/html/router/soft1200/117358AA/B_38.HTM> [retrieved on 2001-10-24] Configuring IP Utilities (117358-A Rev.A) the whole document -----	1-3, 12-15

2